AD-A105 491 UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES ELECTR--ETC F/G 9/2
ANALYSIS AND DESIGN OF FAULT-TOLERANT COMPUTER SYSTEMS.(U)
AUG 81 J P HAYES AFOSR-77-3352
UNCLASSIFIED AFOSR-TR-81-0671 NL

1 OF 1
AD
A105491

END
DATE
FILMED
11-81
DTIC

AFOSR-TR- 81-0671

AD A105491

Final Technical Report

Air Force Office of Scientific Research

Grant No. AFOSR-77-3352

(July 1, 1977 - June 30, 1981)

ANALYSIS AND DESIGN OF FAULT-TOLERANT COMPUTER SYSTEMS

Prepared
by

John P. Hayes

Electronic Sciences Laboratory
University of Southern California
Los Angeles, California 90007

DTIC
ELECTE
OCT 14 1981
A

August 15, 1981

DTIC FILE COPY

Approved for public release;
distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR- 81 -0671	2. GOVT ACCESSION NO.: AD A105 491
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Analysis and Design of Fault-Tolerant Computer Systems	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report, July 1977 - June 1981
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): John P. Hayes	8. CONTRACT OR GRANT NUMBER(s): AFOSR-77-3352
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University of Southern California, Los Angeles, California 90007	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2304/A6 61102F
11. CONTROLLING OFFICE NAME AND ADDRESS: Mathematical & Information Sciences Directorate, Air Force Office of Scientific Research (AFSC), Bolling AFB, D.C. 20332	12. REPORT DATE: 15 August 1981
	13. NUMBER OF PAGES: 25
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS (of this report): Unclassified
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Bit-sliced microprocessors
Connecting networks
Fault diagnosis
Fault-tolerant computing
Microprocessors
Multiprocesssors
Test generation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is the final report of a four-year investigation of fault-tolerant computer systems. A comprehensive theory of fault tolerance for a class of computer interconnection networks called $\mathcal{E}$-networks was developed. Reconfiguration and recovery strategies in multiprocessor systems were studied using graph theoretical models. The testing requirements of bit-sliced computers were investigated, and a new approach to the design of easily testable bit-sliced systems was developed. The testing of complex LSI/VLSI systems was examined using a high-level and expandable representation of test data called vector sequences.

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

This is the final report of a four-year investigation of fault-tolerant computer systems. A comprehensive theory of fault tolerance for a class of computer interconnection networks called ρ-networks was developed. Reconfiguration and recovery strategies in multiprocessor systems were studied using graph theoretical models. The testing requirements of bit-sliced computers were investigated, and a new approach to the design of easily testable bit-sliced systems was developed. The testing of complex LSI/VLSI systems was examined using a high-level and expandable representation of test data called vector sequences.

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DTIC
This technical report has been reviewed and is approved for public release IAW AFR 190-12.
Distribution is unlimited.
MATTHEW J. KERPER
Chief, Technical Information Division

TABLE OF CONTENTS

page

Abstract i

1. Research Objectives 1

2. Research Accomplishments 2

2.1 Fault-Tolerant Interconnection Networks 2

2.2 Analysis of Reconfiguration and Recovery 7

2.3 Bit-Sliced Microprocessor Systems 10

2.4 Testing General LSI/VLSI Systems 14

2.5 References 17

3. Publications 19

4. Personnel 20

5. Interactions 21

1. RESEARCH OBJECTIVES

The purpose of this research project is to develop methods for the analysis and synthesis of complex fault-tolerant computer systems. It is motivated by recent rapid developments in large scale and very large scale integration (LSI and VLSI) technology, especially the introduction of microprocessors and microcomputers, which are expected to increase greatly the need for highly reliable computer systems. The research is concerned with fault diagnosis, reconfiguration and recovery in the event of failures. Its goals include the development of specific measures of the cost and complexity of fault tolerance, and the derivation of efficient fault-tolerant design algorithms based on these measures. The problems associated with the design of systems containing many microcomputers were studied, with emphasis on the connecting networks required for fault-tolerant intercomputer communication. Testing procedures and easily-testable design methods for complex digital systems employing LSI/VLSI technology were also investigated.

2. RESEARCH ACCOMPLISHMENTS

Major new results were obtained in the following areas:

(1) Fault-tolerant interconnection networks

(2) Analysis of reconfiguration and recovery

(3) Bit-sliced microprocessor systems

(4) Testing general LSI/VLSI systems

These results are summarized in this section; detailed descriptions can be found in the cited references.

2.1 Fault-Tolerant Interconnection Networks [1-2]

A comprehensive study of the fault-tolerance requirements of complex multicomputer systems, such as systems containing large numbers of microprocessors, was completed. This work is fully documented in John P. Shen's Ph.D. Dissertation [2]. A survey of the interconnection requirements of multicomputer systems was carried out which lead to the conclusion that a class of interconnecting networks called β-networks constituted one of the most practical communication structures for such systems. Although the communication requirements of β-networks have been studied in the past, mainly in the context of telephone switching systems [3,4], their fault tolerance properties have received little attention. It was realized early in this study that interconnecting networks play a critical role in the reliability and fault tolerance of computer systems.

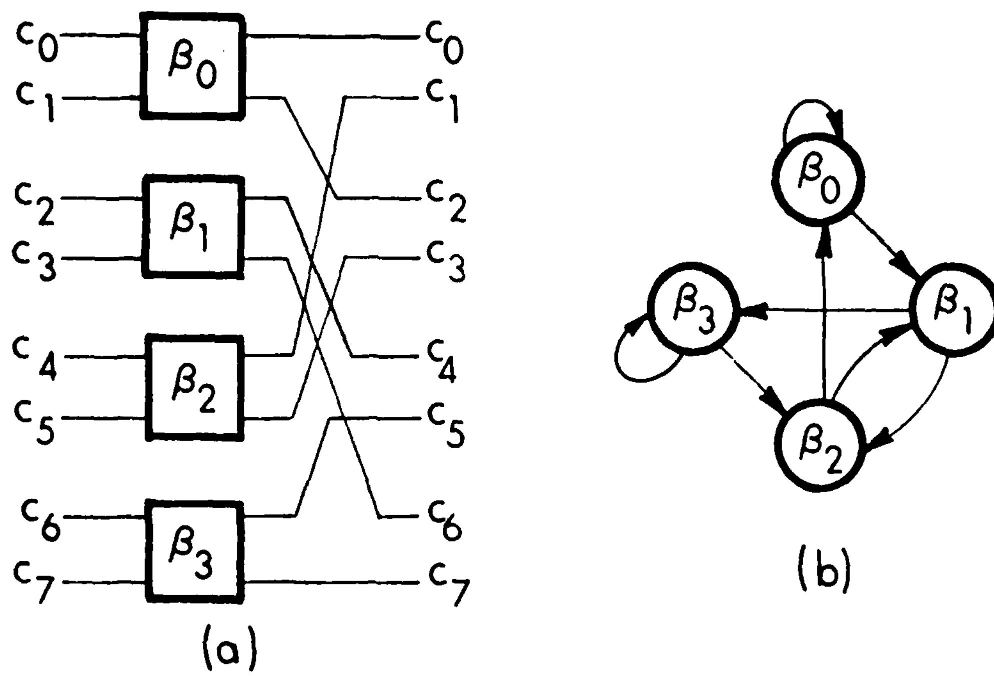


Fig. 1 (a) A β-network and (b) the corresponding β-graph

A β-network is a connecting network composed of 2x2 crossbar switches. Fig. 1a shows a simple network composed of four β-networks which can provide communicated between the eight computers denoted C_0, C_1, ..., C_7. We have introduced an analytical model called a β-graph which allows a β-network to be represented by a standard directed graph. Fig. 1b shows the β-graph that represents the β-network of Fig. 1a. Many of our results are expressed in terms of β-graphs or other graphs derived from β-graphs.

In this analysis we assume that faults in β-networks are caused by failure of the individual β-elements. A β-element has two states during normal operation, a through (T) and a cross (X) state. A fault may cause β-elements to become stuck in either the T or the X state. We have developed a new measure of the fault tolerance of β-networks using a connectivity criterion called dynamic full access (DFA). A β-network is said to have DFA if each of its inputs can be connected to any of its outputs by means of a finite number of passes through the network. Note that the computers provide a set of feedback paths that allow information to be routed from computer to computer until the desired destination is reached. Fault tolerance is achieved by rerouting data transmissions to avoid faulty β-elements or computers.

A fault in a β-network is called critical if it destroys DFA. A minimal critical fault (MCF) is one none of whose subsets is a critical fault. We have obtained several complete graph-theoretical characterizations of the critical faults of a β-network; for details see [1, 2]. For example, we have shown that a fault f is critical if and only if the state due to f is not compatible with any state of the β-network that creates an Eulerian circuit, i.e., a single closed path through all edges, in the corresponding β-graph.

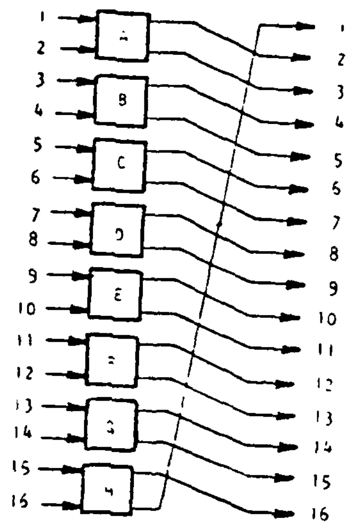
A β-network is defined to be k-fault tolerant or k-FT is the failure of any k or fewer β-elements does not destroy DFA. The largest k for which a β-network is k-FT is called the fault tolerance (FT) parameter of the β-network. In the synthesis of practical fault tolerant β-networks, network performance must also be considered. A performance criterion called the communication delay (CD) parameter was introduced, which is defined as the worst posible transmission delay through the β-network in terms of the number of intervening β-elements between any pair of communicating devices. It was proven that the FT parameter k and CD parameter d of any β-network with n β-elements must satisfy the following fundamental bounds:

$$0 \leq k \leq n-1$$

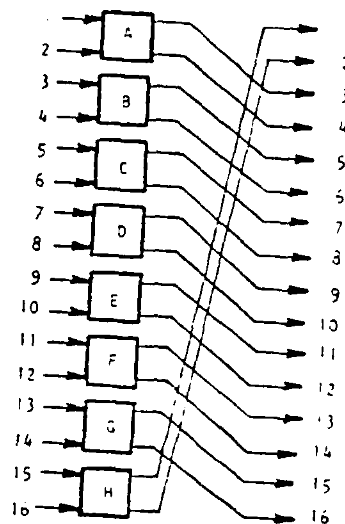
$$\lfloor \log_2 n \rfloor + 1 \leq d \leq n.$$

It has also been shown that these bounds are tight.

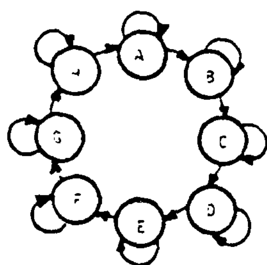
The design of fault tolerant β-networks for multicomputer systems typically involves striking a balance between fault tolerance and communication delay. Two β-network designs were obtained which possess extreme values for k and d. The modified inverse shuffle-exchange (MISE) β-network was shown to have FT parameter $k = 1$ and CD parameter $d = \lfloor \log_2 n \rfloor + 1$. Another β-network called the double parallel ring (DPR) network was shown to have FT parameter $k = n-1$ and CD parameter $d = n$. It was further demonstrated that the DPR-network is unique in achieving the maximum value n-1 of the FT parameter. These results shed considerable light on aspects of β-network behavior which are often not obvious from the network structure alone. For example, Fig. 2 shows two 16x16 β-networks of very similar structure. However, their FT and CD parameters are quite different. The network of Fig. 2a is 0-FT, as is



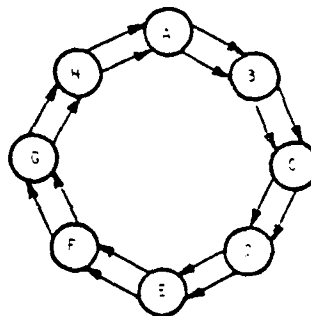
(a)



(c)



(b)



(d)

Fig. 2 (a) A 0-FT β-network, and (b) its β-graph.

(c) A 7-FT DPP network, and (d) its β-graph.

evident from its β-graph appearing in Fig. 2b. The β-network of Fig. 2c is the DPR-network with n=8, hence it is 7-FT. It can easily be seen from the corresponding β-graph in Fig. 2d, that the CD parameter of this β-network is seven.

The preceding theoretical results were applied to the analysis of various β-network structures. Some new properties of cascaded β-networks were derived. The FT and CD parameters were obtained for each of the following well-known β-networks: the double-tree (DOT) network, the indirect binary m-cube (m-IBC) network, and the Benes rearrangeable (BRS) network.

2.2 Analysis of Reconfiguration and Recovery [5]

Most previous research in fault-tolerant computer design has been concerned either with system reliability or fault diagnosis. Other important aspects of system behavior, notably recovery, have received little attention, even though they play a central role in fault tolerance. In this project a new method for analyzing recovery in fault-tolerant multiprocessor systems was developed. The system is represented by a redundant facility graph G_r in which nodes correspond to processors and edges correspond to communication links [6]. The fault-free nodes include nodes actively engaged in data processing and nodes acting as standby spares. A fault is represented by the removal of a node and its associated edges from G_r. Faults are tolerated by reconfiguring the active and spare nodes in G_r so that there always exists an active subnetwork that is isomorphic, that is, has the same interconnection structure, as a certain minimum configuration G_b called the basic system. G_b can be taken as the minimum fault-free system needed to perform a particular set of tasks.

A system G_r is called k-fault-tolerant (K-FT) t-step recoverable (t-SR) if it can recover from up to k faults by changing the states of at most t fault-free nodes. k is clearly a measure of the amount of damage the system can tolerate. A state change, e.g. from spare to active, typically involves the creation of new logical paths in the system, and the transfer of status information between the affected nodes. If t state changes are required to recover from a particular fault, then t is an approximate measure of the system's recovery time. Clearly t is at least equal to k.

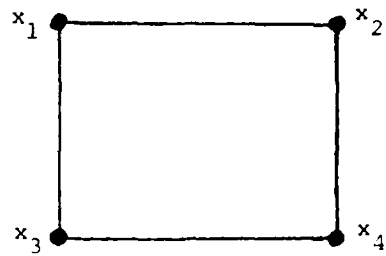
A case of particular interest, corresponding to a class of systems with minimum recovery time, has t = k. In such systems recovery from t faults is achieved by immediate replacement of each failed node by a fault-free spare. G_r is defined to be optimally t-SR with respect to an n-node *basic system* G_b if

(1) G_r is t-FT/t-SR with respect to G_b

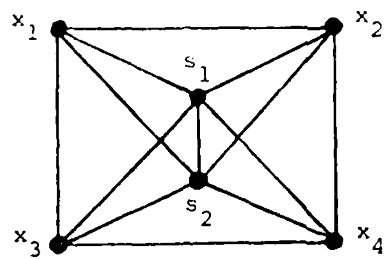
(2) G_r contains the minimum number of nodes, viz. $n + t$

(3) G_r contains the fewest edges among all systems satisfying conditions (1) and (2)

We have shown that the optimal t-SR realization of every G_b is unique, and that it has a relatively simple structure [5]. Figure 3a shows an example of a basic graph I_b consisting of four processors arranged in a ring. Figure 3b shows the corresponding optimal 2-SR graph I_2^{OPT}. It consist of I_b with two additional spare nodes, labeled s_1 and s_2, and additional edges connecting s_1 and s_2 to all nodes of I_b^{OPT}. Every fault graph formed by removing one or two nodes from I_2^{OPT} contains a subgraph isomorphic to I_2 (the 2-FT property).



(a)



(b)

Figure 3. (a) A 4-node basic system I_b.

(b) The corresponding optimal 2-SR system I_b^{OPT}.

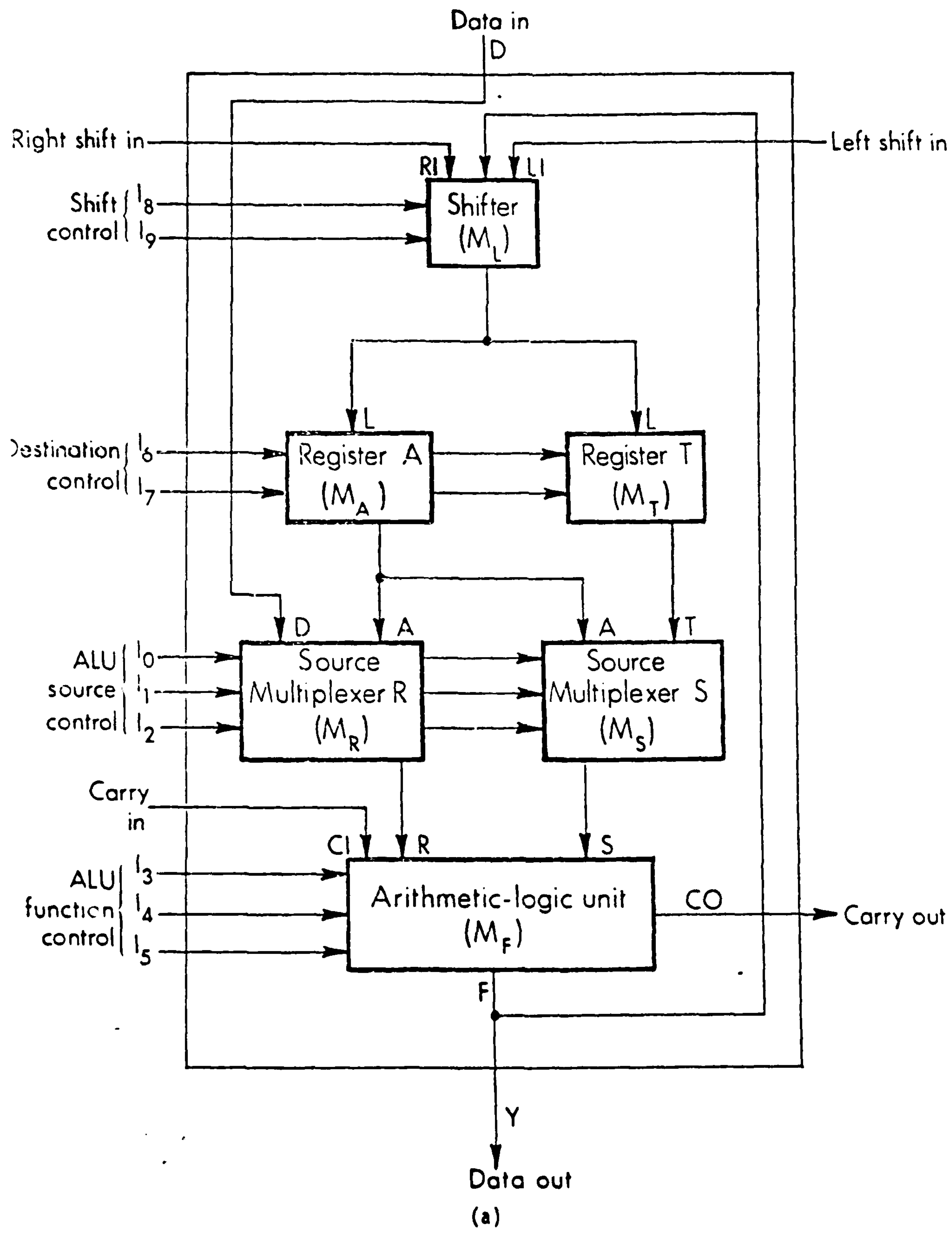
Furthermore, each such subgraph can be chosen so that it differs from the original active subgraph in at most two nodes (the 2-SR property).

Optimal t-SR systems have the disadvantage that the number of edges connected to some nodes i.e., the node degree, may be very large. Since this represents the number of parallel data paths to a processor, it is often severly restricted by physical considerations, for example, microprocessor pin limitations. Thus nonoptimal fault-tolerant systems with limited node fanout are of interest. We have investigated a class of graph transformations, called line graph transformations, which lead to t-SR designs with nodes of lower degree than the corresponding optimal t-SR systems [5]. We have also shown that line graph transformations greatly simplify the computation of the parameters k and t.

2.3 Bit-Sliced Microprocessor Systems [7-12]

A major investigation of the testing requirements of bit-sliced computers has been completed under partial AFOSR sponsorship [7-12]. Bit-sliced systems are an important class of digital systems that have a regular array-like structure, which is particularly attractive for VLSI technology. A bit-sliced system is realized by interconnecting identical slices or cells to form a one-dimensional iterative logic array (ILA). In this study the design of bit-sliced systems that are easily testable has been investigated.

First an analytic test generation methodology for bit-sliced and related systems was developed. For this purpose a high-level (register-level) circuit model and a corresponding functional fault model were specified. A 1-bit processor slice C having the main features of commercial slices was defined as a test case. Figure 4 shows the internal organization of C. Using the high-level circuit and fault models a technique for deriving a complete and near-minimal



(a)

Fig. 4 1-bit processor model C

length test sequence for C was developed. The cell C was then extended to form general k-bit slices whose internal structures more closely resemble commercial products. It was shown that test patterns for an array of N identical processor cells can be easily derived from the tests for an individual cell. Furthermore, the number of test patterns needed for the processor array is independent of the array length. It was therefore concluded that for test generation purposes, bit-sliced processors can be usefully modeled as C-testable ILAs, which require a constant number of test patterns independent of array size.

The property of C-testability in one-dimensional ILAs was studied in detail [9, 10]. Basic concepts of C-testability in unilateral combinational arrays were investigated first. C-testable arrays were characterized and procedures to construct test patterns for such arrays were developed. A new design method to make an ILA C-testable was proposed. C-testable arrays of bilateral and sequential cells were also analyzed. A characterization of C-testable bilateral combinational ILAs was obtained, as well as a design modification scheme to make a bilateral ILA C-testable. It was shown that the results on C-testable combinational arrays can be applied directly to a useful class of sequential arrays.

A new testability criterion for ILAs called I-testability was introduced [8, 10, 11]. I-testability ensures that identical test responses can be obtained from every cell of an ILA, and thus simplifies response verification. I-testable combinational ILAs were characterized, as well as CI-testable ILAs that are simultaneously C- and I-testable. A design scheme for making an arbitrary ILA CI-testable was also constructed. Finally, a design technique for realizing self-testing bit-sliced computers based on I-testing was developed. It was

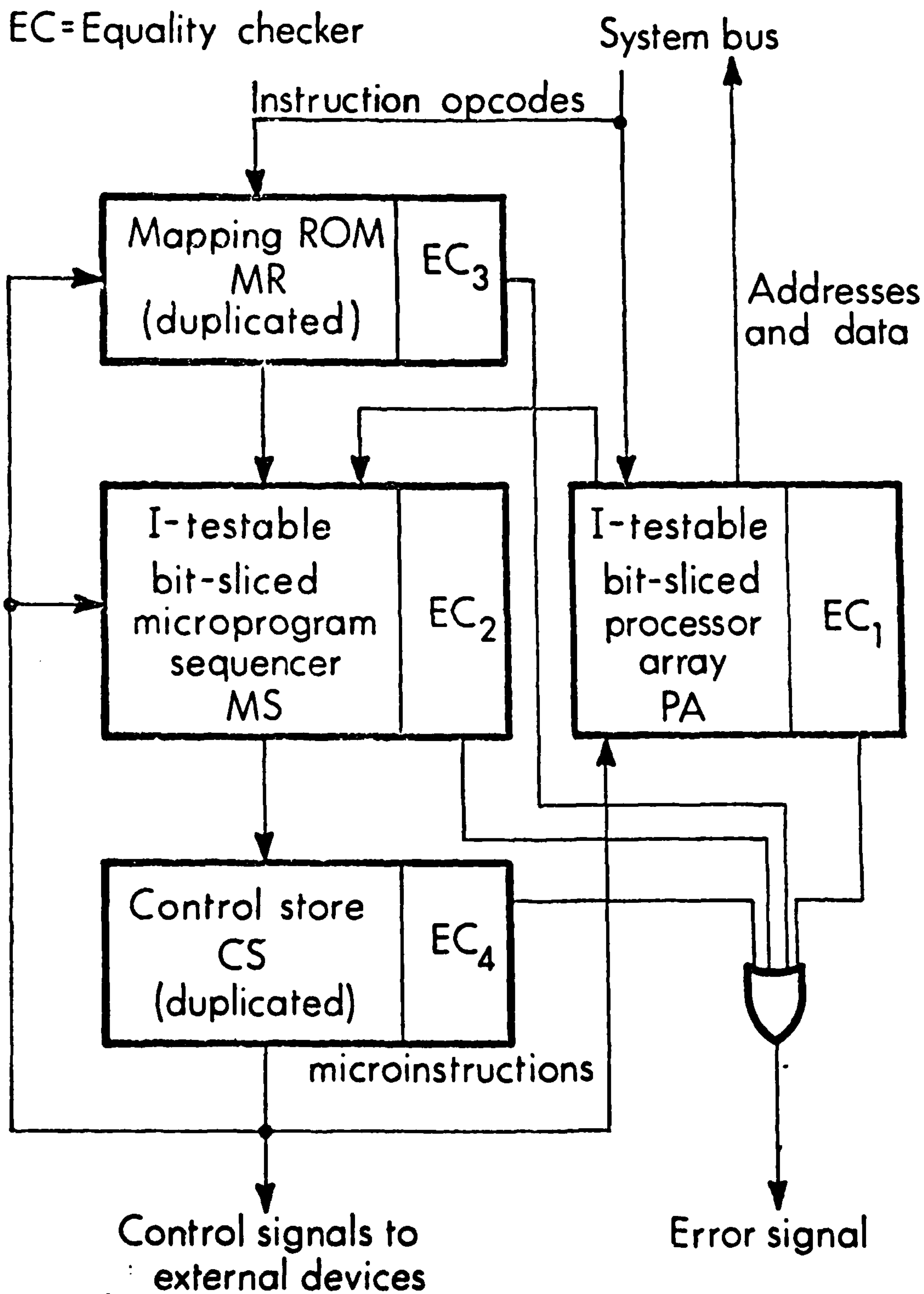


Fig. 5. A self-testing bit-sliced CPU.

established that the family of processor arrays constructed of cell C and its extensions is CI-testable. Using CI-testable processor arrays and other I-testable bit-sliced circuits, a self-testing computer was designed. Figure 5 shows the central processing unit (CPU) of this computer. The advantages and limitations of the proposed design were analyzed and compared to more conventional self-checking approaches that are based on coding techniques [10, 11].

In conjunction with a VLSI design course at USC in Spring 1980, we carried out the complete IC chip design of a 4-bit microprocessor composed of four copies of the slice C [11]. This design was done using the software design tools developed at Caltech and Xerox Corporation [16]. The resulting chip was fabricated using NMOS technology in Summer 1980 as part of a multi-university VLSI project sponsored by ARPA and Xerox. Considerable insight into the problems of VLSI design were obtained from this work, as well as a much better understanding of the limitations imposed by IC technology on the testability of complex computer circuits.

2.4 Testing General LSI/VLSI Systems [13-15]

Most existing analytical tools are inadequate for dealing with digital components above the gate and flip-flop levels which correspond to small-scale integration (SSI) in current technology. There is at present no adequate theory for the design or testing of LSI/VLSI devices, although the need for such a theory has long been recognized [15].

We have observed that a significant property of components at all complexity levels is expansibility, which is the ability of components of a given type to be interconnected in a systematic way to form larger components of the same type [13]. The larger component performs the same operation as its con-

stituent elements, but processes more and/or bigger operands. Expansibility plays a particularly important role in the architecture of microcomputers. The major design problems revolve around the number, size and interconnections of the ROM's, RAM's and IO interface circuits used, problems which are intimately associated with the expansibility of these components. With bit-slice architecture, the CPU (microprocessor) becomes an expandable design component. Two main expansion techniques have been identified, expansion by composition and by replication. It has been demonstrated that expansion methods, can be concisely defined by recursive logic equations. We have shown that most standard components can be expanded using sets of recursive equations called FS2 algorithms which allow neither feedback nor constant input/output values, and which require at most two logic levels. Several other useful expansion methods have also been identified [13].

A new approach to processing the very large amounts of test and response data associated with very complex digital systems such as VLSI circuits was developed [14]. This approach treats complex digital signals, called vector sequences, as primitive elements; this implies that complex subcircuits can also be treated as primitive. New operators for manipulating vector sequences were discovered, and their basic properties were investigated. We have shown that substantial compression of test information is achievable using the vector sequence approach.

The elements of this testing approach are sequences of digital signals appearing in the input/output lines of logic components or circuits. Such a sequence S is represented by a 2-dimensional matrix which is called a *vector sequence*. For example, the binary test sequence applied to an 5-input circuit in six clock periods might be denoted by the following 5x6 vector sequence

$$S_{in} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (1)$$

The horizontal dimension of this matrix represents time suitable quantized, while each position in the vertical or space dimension corresponds to a distinct line or bus L in the circuit under consideration. The quantization of the vertical dimension corresponds directly to the complexity level of the information units and circuits components being used. Any submatrix of a vector sequence can be represented by a primitive symbol. For example, if

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

then we can replace (1) by

$$S_{in} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad (2)$$

In (2) S_{in} is represented by a 2x2 matrix, which corresponds to a higher-level view of S_{in} than (1). The highest-level occurs when S is treated as a 1x1 vector sequence, that is, as a single primitive signal.

We have defined a set of fundamental operators, denoted $\{\cdot, \odot, \times, \otimes\}$ for processing vector sequences. The operators $\cdot$ and $\odot$ represent concatenation (external expansion) in the time and space dimensions, respectively, while x

and $\mathbb{C}$ denote new operations called internal expansion. We have also defined a set of "standard" vector sequences from which the test data, both input patterns and output responses, for a variety of complex circuits can be constructed. We have shown that the vector sequence approach is applicable to logic circuits at all complexity levels, from gate-level circuits to microprocessor-based systems. We are continuing to work on the development of a test generation algorithm, analogous to the D-Algorithm, in which the test data is represented by vector sequences.

2.5 References

1. J.P. Shen, "Fault tolerance of $\mathcal{E}$-networks in interconnected multicomputer systems," Ph.D. Dissertation, Dept. of Electrical Eningeering, U.S.C., August 1981. To be published as U.S.C. Electronic Sciences Lab. Report No. 510.

2. J.P. Shen and J.P. Hayes, "Fault tolerance of a class of connecting networks," *Proc. Seventh Annual Symp. on Computer Architecture*, pp. 61-71, La Baule, France, May 1980.

3. V.E. Benes, *Mathematical Theory of Connecting Networks and Telephone Traffic*, New York, Academic Press, 1965.

4. G.M. Masson, G.C. Gingher and S. Nakamura, "A sampler of circuit switching networks," *Computer*, vol. 12, no. 6, pp. 32 - 48, June 1979.

5. J.P. Hayes and R. Yanney, "Fault recovery in multiprocessor networks," *Digest Eighth Fault-Tolerant Computing Symp.*, pp. 123-128, Toulouse, 1978.

6. J.P. Hayes, "A graph model for fault-tolerant computing systems," *IEEE Trans. Computers*, vol. C-25, pp. 875-884, September 1976.

7. T. Sridhar and J.P. Hayes, "Testing Bit-Sliced Microprocessors," *Proc. 1979 Internat. Symp. on Fault-Tolerant Computing*, pp. 211-218, Madison, Wisconsin, June 1979.

8. T. Sridhar and J.P. Hayes, "Self-Testing Bit-Sliced Microcomputers," *Digest 22nd IEEE Computer Society Conf. (Spring COMPCOM 81)*, San Franciso, pp. 312-316, Feb. 1981.

9. T. Sridhar and J.P. Hayes, "A Functional Approach to Testing Bit-Sliced Microprocessors," *IEEE Trans. on Computers*, August 1981 to appear.

10. T. Sridhar and J.P. Hayes, "Design of Easily Testable Bit-Sliced Systems," *IEEE Trans. on Computers*, Nov. 1981 to appear.

11. T. Sridhar, "Easily Testable Bit-Sliced Digital Systems," Ph.D. Dissertation, Department of Electrical Eng., U.S.C., July 1981.

12. J.P. Hayes, "A survey of bit-sliced computer design," *Journal of Digital Systems*, 1981, to appear.

13. J.P. Hayes, "Component expansion techniques in computer design," *Digital Processes*, vol. 4, no. 4, pp. 295-312, Winter 1978.

14. J.P. Hayes, "A calculus for testing complex digital systems," *Proc. Tenth Fault-Tolerant Computing Symp.*, Kyoto, Japan, pp. 115-120, October 1980.

15. J.P. Hayes and E.J. McCluskey, "Testability Considerations in Microprocessor-Based Design," *Computer* vol. 13, no. 3, pp. 17-26, March 1980.

16. C. Mead and L. Conway, *Introduction to VLSI Systems*, Reading, Mass., Addison-Wesley, 1980.

3. PUBLICATIONS

The following documents were sponsored wholly or in part by Grant No. AFOSR-77-3352.

1. J.P. Hayes and R. Yanney, "Fault recovery in multiprocessor networks," *Digest Eighth Fault-Tolerant Computing Symp.*, pp. 123-128, Toulouse, 1978.

2. J.P. Hayes, "Component expansion techniques in computer design," *Digital Processes*, vol. 4, no. 4, pp. 295-312, Winter 1978.

3. T. Sridhar and J.P. Hayes, "Testing bit-sliced microprocessors," *Digest Ninth Fault-Tolerant Computing Symp.*, pp. 211-218, Madison, Wisconsin, 1979.

4. J.P. Hayes and E.J. McCluskey, "Testability considerations in microprocessors-based design," Computer Systems Laboratory Technical Report No. 179, Stanford University, November 1979.

5. J.P. Hayes and E.J. McCluskey, "Testability considerations in microprocessor-based design," *Computer*, vol. 13, no. 3, pp. 17-26, March 1980.

6. J.P. Shen and J.P. Hayes, "Fault tolerance of a class of connecting networks," *Proc. Seventh Annual Symp. on Computer Architecture*, pp. 61-71, La Baule, France, May 1980.

7. J.P. Hayes, "A calculus for testing complex digital systems," *Proc. Tenth Fault-Tolerant Computing Symp.*, Kyoto, Japan, pp. 115-120, October 1980.

8. T. Sridhar and J.P. Hayes, "A functional approach to testing bit-sliced microprocessors," *IEEE Trans. Computers*, August 1981, to appear.

9. J.P. Hayes, "A survey of bit-sliced computer design," *Journal of Digital Systems*, 1981, to appear.

10. J.P. Shen, "Fault tolerance of β-networks in interconnected computer systems," Ph.D. Dissertation, Dept. of Electrical Engineering, University of Southern California, August 1981. To be published as USC Electronic Sciences Laboratory Report No. 510.

11. J.P. Shen and J.P. Hayes, "Fault tolerance of β-networks in multicomputer systems. Paper in preparation. To be submitted to *IEEE Trans. on Computers*.

4. PERSONNEL

The following people were associated with the research effort reported here.

Principal Investigator

John P. Hayes

Research Assistants

John P. Shen
Thirumalai Sridhar
Raif Yanney

NOTE: T. Sridhar and R. Yanney received no financial support from Grant No. AFOSR-77-3352.

5. INTERACTIONS

Meetings with Air Force Personnel

J.P. Hayes met with Dr. Joseph Bram, AFOSR Directorate of Mathematical and Information Sciences, in Los Angeles on January 30, 1978; in Stanford on February 8, 1979; and in Los Angeles on April 8, 1980. Current progress and future plans for the project being reported here were reviewed.

J.P. Hayes met with Mr. Armand Vito of RADC (ISCA) in Marina Del Rey, California on April 6, 1979 to discuss research topics of mutual interest.

J.P. Hayes visited Rome Air Development Center (RADC), Rome, New York, May 12-13, 1978. He met with Mr. Murray Kesselman (ISCA) who provided him with a detailed overview of Air Force research interests in the areas of computer architecture and fault-tolerant computing. He also met with Lt. Michael Troutman (ISCA) and discussed the Air-Force-sponsored Total System Design (TSD) and Multi-Microprocessor System (MMS) projects. Dr. Hayes has an opportunity to see some of RADC's research facilities, including its QM-1 and STARAN computers.

Visit to Standford University

J.P. Hayes was a Visiting Associate Professor at the Computer Systems Laboratory, Stanford University during the period January 24 - June 6, 1979. He was associated with the Center for Reliable Computation, and interacted with Prof. E.J. McCluskey and his group on problems of mutual interest in the area of fault-tolerant computing.

Attendance at Technical Conferences

J.P. Hayes and R. Yanney attended the 1978 International Symposium on Fault-Tolerant Computing (FTCS-8) in Toulouse, France, June 1978. The paper "Fault recovery in multiprocessor networks" was presented at this conference.

J.P. Hayes attended the IEEE Computer Society International Confernce (COMPCON) in San Franciso, February 1979. J.P. Hayes and R. Yanney attended the 1979 Design Automation Conference in San Diego, June 1979. J.P. Hayes, T. Sridhar and R. Yanney attended the 1979 International Symposium on Fault-Tolerant Computing (FTCS-9) in Madison, Wisconsin, June 1979. The paper "Testing bit-sliced microprocessors" was presented at this conference.

J.P. Hayes attended the Seventh Annual Symposium on Computer Architecture in La Baule, France, May 1980. The paper "Fault tolerance of a class of connecting networks" was presented at this conference. J.P. Hayes also attended the 1980 International Symposium on Fault-Tolerant Computing (FTCS-10) in Kyoto, Japan, Oct. 1980 to present the paper "A calculus for testing complex digital systems."

J.P. Hayes and T. Sridhar attended the IEEE Computer Society International Conference (COMPCON) in San Francisco, February 1981. J.P. Hayes participated in the IEEE Design for Testability Workshop in Vail, Colorado, April 1981. J.P. Hayes and R. Yanney attended the 1981 International Symposium on Fault-Tolerant Computing (FTCS-11) in Portland, Maine, June 1981.

END

DATE FILMED

11-81

DTIC